

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2000-350043

(43)Date of publication of application : 15.12.2000

(51)Int.Cl.

H04N 1/413

H03M 7/40

H04N 7/32

(21)Application number : 11-157726

(71)Applicant : MATSUSHITA GRAPHIC COMMUNICATION
SYSTEMS INC

(22)Date of filing : 04.06.1999

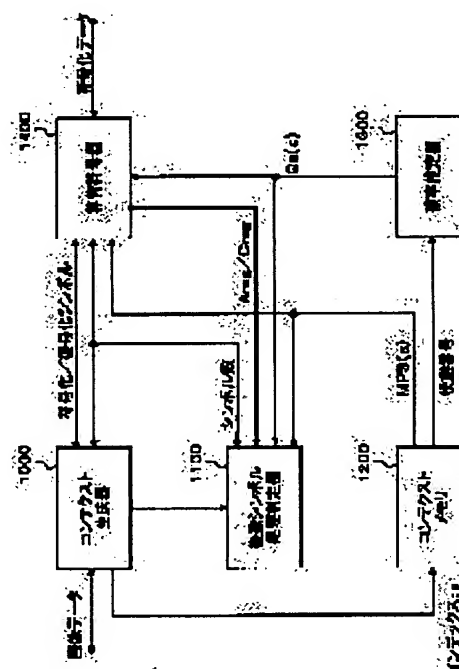
(72)Inventor : HORIE HITOSHI

(54) ARITHMETIC CODER AND ARITHMETIC DECODER

(57)Abstract:

PROBLEM TO BE SOLVED: To attain arithmetic code processing at a high speed, while flexibly coping with complicated state of the image of a coding object.

SOLUTION: A plurality symbol processing discriminator 1100 discriminates whether batch processing of consecutive symbols is available and adaptively executes consecutive coding depending on the situation. This discrimination is conducted using a plurality of detectors, e.g. that are operated in parallel and detecting that consecutive context/superior symbols and normalization processing area not caused to each of consecutive symbol numbers which are an object of batch coding.



LEGAL STATUS

[Date of request for examination] 08.12.2000

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the
examiner's decision of rejection or application converted
registration]

[Date of final disposal for application]

[Patent number] 3350482

[Date of registration] 13.09.2002

[Number of appeal against examiner's decision of
rejection][Date of requesting appeal against examiner's decision of
rejection]

[Date of extinction of right]

Copyright (C); 1998,2003 Japan Patent Office

SPEC

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開2000-350043

(P2000-350043A)

(43) 公開日 平成12年12月15日 (2000. 12. 15)

(51)Int.Cl. ⁷	識別記号	F I	ターマコード*(参考)		
H 0 4 N	1/413	H 0 4 N	1/413	Z	5 C 0 5 9
H 0 3 M	7/40	H 0 3 M	7/40		5 C 0 7 8
H 0 4 N	7/32	H 0 4 N	7/137	Z	5 J 0 6 4

審査請求 未請求 請求項の数20 O L (全 11 頁)

(21) 出願番号 特願平11-157726

(22) 出願日 平成11年6月4日 (1999. 6. 4)

(71) 出願人 000187736

松下電送システム株式会社

東京都目黒区下目黒2丁目3番8号

(72) 発明者 堀江 等

東京都目黒区下目黒2丁目3番8号 松下

電送システム株式会社内

(74) 代理人 100105050

弁理士 鷲田 公一

Fターム(参考) 5C059 KK15 ME11 UA02 UA05 UA38

5C078 BA21 BA32 CA31 DA00 DA01

DA02 DB13

5J064 AA03 BA10 BB12 BC01 BC02

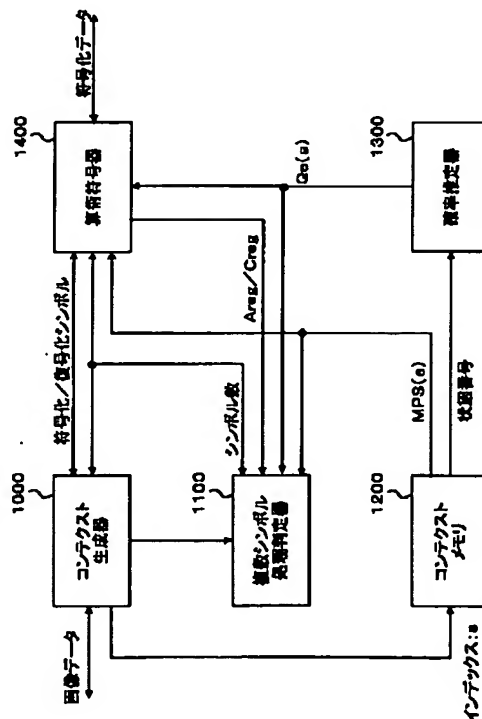
BC14 BC17 BC28 BD01

(54) 【発明の名称】 算術符号化装置および算術復号化装置

(57) 【要約】

【課題】 符号化対象の画像の複雑さに柔軟に対応しつつ、算術符号化処理の高速化を図ること。

【解決手段】 複数シンボル処理判定器1100が、連続したシンボルの一括処理が可能か否かを判定し、状況に応じて適応的に連続符号化を実行する。前述の判定は、例えば、並列動作する複数の検出器を用いて、一括符号化の対象となる連続シンボル数の各々について、コンテキスト/MPSの連続と、正規化処理が発生しないことを検出することにより行われる。



【特許請求の範囲】

【請求項 1】 符号化シンボル周辺の参照画素の値によって符号化シンボルを予測し、その予測結果を符号化する算術符号化装置において、一括符号化するシンボル数を可変としたことを特徴とする算術符号化装置。

【請求項 2】 復号化シンボル周辺の参照画素の値によってシンボルを復号化する算術復号化装置において、一括復号化するシンボル数を可変としたことを特徴とする算術復号化装置。

【請求項 3】 符号化シンボル周辺の参照画素の値によって符号化シンボルを予測し、その予測結果を符号化する算術符号化装置において、
10 予め定めた複数個の連続する符号化シンボルの値および各符号化シンボルを符号化する際に用いられる参照画素の値が所定のパターンに合致することを検出する複数個の検出器と、これらの検出器の検出結果に基づいて、一括して符号化できるシンボル数を決定するシンボル数決定手段と、このシンボル数決定手段によって決定されたシンボル数分の連続したシンボルを一括して符号化する算術符号化手段と、を有することを特徴とする算術符号化装置。

【請求項 4】 前記複数個の検出器はそれぞれ、一括符号化するシンボル数の条件を異にした検出を並列に実行することを特徴とする請求項 3 記載の算術符号化装置。

【請求項 5】 前記シンボル数決定手段は、並列に動作する前記複数個の検出器の中で、前記所定のパターンに合致していることを示す信号を出力しているものがあるかを調べ、複数の検出器が該当する場合には、それらの検出器に割り当てられている一括符号化するシンボル数の条件のうちの最大のシンボル数を選択することを特徴とする請求項 4 記載の算術符号化装置。

【請求項 6】 前記シンボル数決定手段は、一括符号化を行なった場合でも算術符号化手段における正規化処理が発生しないことも条件として、一括して符号化できるシンボル数を決定することを特徴とする請求項 3～請求項 5 のいずれかに記載の算術符号化装置。

【請求項 7】 復号化シンボル周辺の参照画素の値によってシンボルを復号化する復号化装置において、
20 予め定めた複数個の連続する復号化シンボルに対応する参照画素が特定のパターンに合致することを検出する複数個の検出器と、これらの検出器の検出結果に基づいて一括復号化できる最大のシンボル数を決定するシンボル数決定手段と、このシンボル数決定手段によって決定されたシンボル数分の連続したシンボルを一括して復号化する算術復号化手段と、を有することを特徴とする算術復号化装置。

【請求項 8】 前記複数個の検出器はそれぞれ、一括復号化するシンボル数の条件を異にした検出を並列に実行することを特徴とする請求項 7 記載の算術復号化装置。

【請求項 9】 前記シンボル数決定手段は、並列に動作

する前記複数個の検出器の中で、前記所定のパターンに合致していることを示す信号を出力しているものがあるかを調べ、複数の検出器が該当する場合には、それらの検出器に割り当てられている一括復号化するシンボル数の条件のうちの最大のシンボル数を選択することを特徴とする請求項 8 記載の算術復号化装置。

【請求項 10】 前記シンボル数決定手段は、一括復号化を行なっても算術復号化手段において正規化処理が発生しないことも条件として、一括して復号化できるシンボル数を決定することを特徴とする請求項 7～請求項 9 のいずれかに記載の算術復号化装置。

【請求項 11】 符号化シンボル周辺の参照画素の値によって符号化シンボルを予測し、その予測結果を符号化する符号化装置において、
25 予め定めた連続する符号化シンボルの値および各符号化シンボルを符号化する際に用いられる参照画素の値が特定のパターンに合致することを検出する検出器と、この検出器の検出結果に基づいて、所望のシンボル数についての一括符号化が可能であるか否かを判定する判定手段と、この判定手段の判定結果に基づいて一括符号化するシンボル数を変化させる一括符号化シンボル数変更回路と、を有することを特徴とする算術符号化装置。

【請求項 12】 前記判定手段によって所望の一括符号化が不可と判定された場合には、前記一括符号化シンボル数変更回路は、次回の符号化に際して、一括符号化を希望するシンボル数を今回のシンボル数以下の数に変更し、前記判定手段によって所望の一括符号化が可能と判定された場合には、次回の符号化に際して、一括符号化を希望するシンボル数を今回のシンボル数以上の数に変更することを特徴とする請求項 11 記載の算術符号化装置。

【請求項 13】 復号化シンボル周辺の参照画素の値によってシンボルを復号化する復号化装置において、
30 予め定めた連続する復号化シンボルに対応する参照画素が特定のパターンに合致するかを検出する検出器と、この検出器の検出結果に基づいて、所望のシンボル数についての一括復号化が可能であるか否かを判定する判定手段と、この判定手段の判定結果に基づいて一括復号化するシンボル数を変化させる一括復号化シンボル数変更回路と、を有することを特徴とする算術復号化装置。

【請求項 14】 前記判定手段によって所望の一括復号化が不可と判定された場合には、前記一括復号化シンボル数変更回路は、次回の復号化に際して、一括復号化を希望するシンボル数を今回のシンボル数以下の数に変更し、前記判定手段によって所望の一括復号化が可能と判定された場合には、次回の復号化に際して、一括復号化を希望するシンボル数を今回のシンボル数以上の数に変更することを特徴とする請求項 13 記載の算術復号化装置。

【請求項 15】 請求項 1、請求項 3、請求項 4、請求

項5、請求項6、請求項11、請求項12のいずれかに記載の算術符号化装置を有する通信装置。

【請求項16】 請求項2、請求項7、請求項8、請求項9、請求項10、請求項13、請求項14のいずれかに記載の算術復号化装置を有する通信装置。

【請求項17】 コンテキストおよび優勢シンボル(MPS)が連続するという第1の条件と、前記連続する優勢シンボル数分の劣勢シンボル(LPS)の領域幅を一括して現状のオージェントから減算した後に残存する領域の値が所定値未満にならないという第2の条件とを共に満たすかを常に検出し、それらの条件が満たされる場合には、前記条件を満たす最大のシンボル数分の一括符号化を適応的に実行することを特徴とする算術符号化方法。

【請求項18】 コンテキストおよび優勢シンボル(MPS)が連続するという第1の条件と、前記連続する優勢シンボル数分の劣勢シンボル(LPS)の領域幅を一括して現状のオージェントから減算した後に残存する領域の値が所定値未満にならないという第2の条件とを共に満たすかを常に検出し、それらの条件が満たされる場合には、前記条件を満たす最大のシンボル数分の一括復号化を適応的に実行することを特徴とする算術復号化方法。

【請求項19】 一括符号化を希望するシンボル数を設定し、その設定されたシンボル数について、コンテキストおよび優勢シンボル(MPS)が連続するという第1の条件と、前記連続する優勢シンボル数分の劣勢シンボル(LPS)の領域幅を一括して現状のオージェントから減算した後に残存する領域の値が所定値未満にならないという第2の条件とを共に満たすかを検出し、それらの条件が満たされる場合には、前記設定されたシンボル数分の一括符号化を実行するとともに、次の符号化における一括符号化を希望するシンボル数として、今回の希望するシンボル数以上のシンボル数を設定することを特徴とする算術符号化方法。

【請求項20】 一括復号化を希望するシンボル数を設定し、その設定されたシンボル数について、コンテキストおよび優勢シンボル(MPS)が連続するという第1の条件と、前記連続する優勢シンボル数分の劣勢シンボル(LPS)の領域幅を一括して現状のオージェントから減算した後に残存する領域の値が所定値未満にならないという第2の条件とを共に満たすかを検出し、それらの条件が満たされる場合には、前記設定されたシンボル数分の一括復号化を実行するとともに、次の復号化における一括復号化を希望するシンボル数として、今回の希望するシンボル数以上のシンボル数を設定することを特徴とする算術復号化方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、算術符号化装置お

および算術復号化装置に関する。

【0002】

【従来の技術】算術符号化において、2値画像をマルチ情報源でモデル化し、符号化シンボルをその周辺画素の状態によって予測し、予測結果を算術符号化する方式が圧縮率の点から最も優れた特性を示すことが知られている。

【0003】JBIG(ITU勧告T.82)に採用されているQM-coderがその例である。しかし、このような方式は圧縮率は優れているが、1シンボル毎にコンテキストの生成、シンボルの生起確率情報の推定、算術符号化演算を繰り返すため処理時間が大きいという問題がある。

【0004】図13にQM-coderの主要な構成を示す。

【0005】図示されるように、QM-coderは、コンテキスト生成部200と、コンテキストテーブル(RAM)210と、確率推定部220と、算術符号器230とで構成されており、次のように動作する。

【0006】まず、コンテキスト生成部200において、符号化画素の周辺10画素によって作られる1024個の状態を検出する。各状態をコンテキスト(s)と呼び、コンテキスト毎に優勢シンボルの予測値MPS(s)と確率推定器の状態番号がコンテキストテーブル210から読み出され、確率推定部220に出力される。確率推定部は、これらの情報から領域幅 $Qe(s)$ を算術符号器に出力する。算術符号器230は、符号化シンボル値と、優勢シンボルの予測値MPS(s)と、劣勢シンボルの生成確率に対応する領域幅 $Qe(s)$ とから算術符号化演算を実行する。

【0007】次に、算術符号演算について図14を参照して説明する。

【0008】算術符号化では、初期値0~1の数直線を優勢シンボル(MPS)の領域幅と劣勢シンボル(LPS)の領域幅に分ける。QM-coderでは、数直線の上側にLPS幅、下側にMPS幅が割り当てられている。2つの領域幅を足した幅をオージェントとよび、以下Aregと表す。

【0009】図14は、“0100”に対する領域分割の様子を示している。図中、A(X)はシンボル系列Xに対応した領域の幅を表す。例えば、部分区間A(0100)は、シンボル系列“0100”及び生成確率に対応する。符号化対象シンボル系列は、分割された領域内の代表点に対応させる。代表点は、部分区間内の一番下にとられる。

【0010】符号化シンボルと予測値が同じ時は、次のシンボルの符号化にはMPS幅が選ばれ、そうでなければLPS幅が選ばれる。上述のとおり、この領域幅の中に代表点を設けて、その2進少数点で符号を表す。

【0011】算術符号化演算では、領域幅が「所定値」未満になった時には、小数点の精度の低下を防ぐために、所定値以上になるまで2倍処理を繰り返す(この処理を正規化処理という)。ここで、「所定値」は初期値の1/2である。

【0012】正規化処理はLPSを符号化したときにも行

5

われる。正規化処理では確率推定部の状態遷移が更新され、次のシンボルから $Q_e(s)$ の値が異なり、これによって、より情報源の確率分布に適した値が選択されるようになる。

【0013】MPS幅は大きく、LPS幅 $Q_e(s)$ はMPS幅よりも小さな値が割り当てられている。MPSの符号化では、1シンボルの符号化後のオーギュメント(Areg)の値は、符号化前のオーギュメント(Areg)からLPS幅 $Q_e(s)$ を減算した値(すなわち、 $Areg - Q_e(s)$)となる。この値が初期値の1/2になるまで正規化処理は行われない。

【0014】コンテキストが同一でMPSが連続するときには、各シンボルの予測条件には変化がないため、 $Areg - n \times Q_e(s)$ のように予め定めた n シンボル分、一括して演算することができる。このような一括符号化方式は、特公平7-95693(符号化復号化装置)に開示されている。

【0015】

【発明が解決しようとする課題】しかし、上記の従来方式は予め定めたシンボル数に対して一括処理の条件が満たされた場合に限って高速化が可能である。このような条件を満たす可能性のある画像パターンは画素配置の水平移動によってコンテキストが同一となるので、参照画素とMPSが全て白、全て黒、および横方向のストライプに限られる。現実的には、横方向のストライプパターンや全黒パターンの発生頻度は低いので、全白部分で効果があると考えられる。

【0016】しかし、画像の中の全白部分は文字間、行間、左右の余白などであり、解像度や画像の種類によって全白部分の大きさは大きく変化する。また、誤差拡散画像では、一括処理シンボル数を大きくすると、一括処理条件を満足する頻度は極めて小さくなり、高速化の効果はほとんど認められなくなる。

【0017】本発明は、このような考察に基づいてなされたものであり、各種の画像や画像の局所状態に柔軟に対応して高速処理を行なうことができる算術符号化装置・算術復号化装置を提供することを目的とする。

【0018】

【課題を解決するための手段】本発明は、画像の局所局所の状況に適応して連続処理シンボル数を可変とし、画像の複雑さに応じて高速化の効果が得られるように構成したものである。

【0019】例えば、複数個の検出器で状況を判定し、その状況に応じて適応的に一括符号化のシンボル数を決定するように構成する。

【0020】また、画像の種類や局所状態に適応するよう、連続処理シンボル数を状態遷移によって可変とした構成とする。

【0021】

【発明の実施の形態】本発明の一態様では、一括符号化・復号化するシンボル数を可変とする。

6

【0022】本発明の他の態様では、複数個の検出器によるパターン判定を並列に行ない、各検出器の検出結果に基づいて、一括して符号化・復号化できるシンボル数を適応的に決定する。

【0023】本発明の他の態様では、一括符号化・復号化のシンボル数の決定は、各検出器に割り当てられている一括符号化・復号化するシンボル数の条件のうちの最大のシンボル数を選択することにより行われる。シンボル数の決定にあたっては、算術符号器における正規化処理が発生しないことも条件とするのが望ましい。

【0024】また、本発明の他の態様では、一括符号化を希望するシンボル数を設定し、その設定されたシンボル数についての一括符号化が可能であるか否かを判定手段により判定し、その可否に応じて、次の符号化・復号化におけるシンボル数(設定値)を変更する。すなわち、判定結果が「可」であれば、次のシンボル数を今回のもの以上とし、「否」であれば、今回のもの以下とする。

【0025】本発明によれば、比較的簡単で現実的な構成でもって、適応的な一括の算術符号化・復号化を行なうことができ、符号化対象の複雑さに対応して、可能な限りの高速処理が常に行なわれる。よって、符号化・復号化処理の効率が向上する。ファクシミリ装置のような通信装置に搭載すれば、文字画像のみならず中間調画像(網点画像)についても適切な一括符号化・復号化処理を行なえる、高機能な通信装置が実現される。

【0026】(実施の形態1)図1は本発明による算術符号・復号器の全体構成を示すブロック図である。

【0027】基本的な構成は図13と同様であるが、複数シンボル処理判定器1100を有するのが特徴である。図示されるように、構成要素は、コンテキスト生成器1000、複数シンボル処理判定器1100、コンテキストメモリ1200、確率推定器1300および算術符号器1400である。

【0028】画像データはコンテキスト生成器1000に入力され、そこで定められた配置の参照画素の値によって、インデックス(s)を出力する。コンテキスト生成器1000からは更に、複数シンボル処理判定器1100に対して符号化シンボルの周辺画素データが出力される。

【0029】コンテキストメモリ1200は、1024バイトのRAMで構成されている。一つのコンテキストに1バイトが割り当てられ、各バイト内には1ビットで表された符号化シンボルの予測値MPS(s)と確率推定器1300の状態番号7ビットとが記憶されている。

【0030】確率推定器1300はROMで構成された状態遷移テーブルである。確率推定器1300はLPSの領域幅 $Q_e(s)$ を算術符号器1400に出力する。算術符号器1400は、符号化シンボル、MPS(s)、 $Q_e(s)$ および複数シンボル処理判定器1100の出力信号である連続処理シンボル数を用いて、符号データ系列を出力する。

【0031】また、複数シンボル処理判定器1100は、符

号化シンボルの周辺画素データと、算術符号器のAレジスタ値AregおよびQe(s)の値を用いて、そのシンボルから予め定めた範囲内で連続処理可能なシンボル数の最大値を算出する。

【0032】本実施例では、並列に動作する複数の検出器を設け、コンテキストとMPSが連続するかを、2シンボル、4シンボル、8シンボル、16シンボルについて並行して検出し、その最大値を選択する。

【0033】例えば、「2シンボル」と「4シンボル」の検出器が共に一括処理可能と判定した場合には、「4シンボル」が一括処理のシンボル数として選ばれる。但し、「4シンボル」の一括符号化をすると初期のオージェント(Areg)の中央値(半分の値)をきってしまったて正規化処理が生じる場合には、処理の複雑化を避けるために、「2シンボル」を選択することになる(2シンボルでも正規化処理が発生する場合には逐次処理となる)。

【0034】すなわち、コンテキストが同一でMPSが連続するときには、各シンボルの予測条件には変化がないため、 $Areg - n \times Qe(s)$ のように予め定めたnシンボル分、一括して演算することができるのであるが、図2に示すようにしきい値(B点, "0x8000")を下回るような場合には、一括処理を禁止する。

【0035】したがって、図3に示すように、一括処理が可能か否かの判定としては、正規化処理が発生するかの判定(ステップ10)と、コンテキストとMPSが連続するかのパターン判定(ステップ20)の2種類の判定がある。そして、各条件が共に満たされる場合に一括処理が行われ(ステップ30)、それ以外は逐次処理(ステップ40)となる。

【0036】以下、図1に示される各ブロックの構成や動作について具体的に説明する。

【0037】図4は、コンテキスト生成器1000のブロック図である。図示されるように、コンテキスト生成器1000は、3本のシフトレジスタ1010~1030と、インデックス生成器1040と、全体を制御するタイミング制御回路1050とを有する。

【0038】シフトレジスタは4バイトで構成され、符号化/復号化ライン用のシフトレジスタ1010と、参照ライン用のシフトレジスタ1020、1030に大別される。符号化のときは、画像データは、メモリ(図示せず)から読み出され、各シフトレジスタの最下位バイトから順次、入力される。

【0039】なお、シフトレジスタのビット位置は、図示したように、c-7~c23, b-7~b23, a-7~a23の記号で識別することにする。

【0040】次に、符号化時の動作を説明する。

【0041】初期状態で全てのシフトレジスタはクリアされる。そして、画像データがメモリ(図示せず)から読み出され、符号/復号化ライン用のシフトレジスタ10

10の最下位バイトから順次、入力される。その後、符号化対象のライン第1画素がa0の位置にくるまで左にシフトし、更に下位バイトをメモリから読み出し3バイトの画像データを詰める。参照ラインについても同様に、ラインの第1画素がb0, c0に来るようにする。

【0042】データ格納が完了した段階で、a0の位置にあるのが符号化シンボル(図中、"?"で示されている)である。また、a-2, a-1, b-2, b-1, b0, b1, b2, c-1, c0の各位置にある10画素が参照画素である。なお、これらのシンボルは、図4中で、太い点線で囲まれている。

【0043】また、インデックス生成器1040は、参照画素をa-2, a-1, b-2, b-1, b0, b1, b2, c-1, c0の順に並べる。この10ビットのデータが、コンテキストを識別するインデックス(s)となる。

【0044】符号化シンボル"?"を符号化した後は逐次処理であれば、シフトレジスタ全体を左に1ビットシフトして新しいコンテキストを作る。

【0045】復号化のときは、復号シンボルはa0の位置に書き込まれる。1シンボル毎の逐次処理では1シンボル復号する毎にシフトレジスタを左に1ビットシフトする。復号データは8ビット毎にメモリに書き込む。このデータはシフトレジスタのa-7~a-1のバイトデータである。

【0046】画像データの入出力、ビットシフトなど全体の動作タイミングはタイミング制御回路1050によって制御される。以上がコンテキスト生成器1000の構成と動作である。

【0047】次に、複数シンボル処理判定器1100の構成と動作について説明する。

【0048】一括(連続)の符号化処理を行なう場合には、コンテキスト(参照画素の状態)が所定のパターンであること(例えば、全部の画素が白である状態が続いていること)と、符号化シンボルについての優勢シンボル値(MPS)が所定パターンであること(つまり、連続していること)の双方を判定する必要がある。MPSの連続も判定するのは、参照画素が同じであっても、符号化対象の画素の値によってMPSが変化し、この場合には、一括した処理ができないからである。図4で説明したコンテキスト生成器1000に蓄積された画像データは、同一コンテキストとMPSが連続しているかどうかを判定するために、複数シンボル処理判定器1100に出力される。ここで、出力される画像データは、図4のa-2~a15, b-2~b17, c-1~c16の各ビット位置のデータであるものとする。

【0049】図5は、複数シンボル処理判定器1100の構成を示すブロック図である。

【0050】先に説明したように、本実施例では連続処理可能な複数シンボルの候補として、「2シンボル」、「4シンボル」、「8シンボル」および「16シンボル」

の4つを設定しており、実行可能なものの中から最大値を選ぶ。

【0051】すなわち、2、4、8、16の各連続シンボルのうち、実行可能なものはどれかを判定するために必要な情報を、各連続シンボルに対応して設けられた複数の検出器の各々で取得し、その情報を連続処理シンボル数判定回路1113に入力する構成である。

【0052】複数の検出器の各々は、図3で示した2種類の判定を行なうための構成を有しており、それぞれ同じ構成をもつ。ここでは、「16シンボル」についての検出を行なう場合について説明する

16シンボルの一括処理に対応した検出器は、4ビットシフタ1104、減算器1108、比較器1112、比較器1131と、コンテキスト／MPSの16連続判定回路1117と、を有している。

【0053】4ビットシフタ1104、減算器1108、比較器1112、比較器1131は、一括の符号化処理を行なった場合に正規化処理が必要となるかを判断するための判定情報を取得する部分を構成する。なお、比較器1131は復号化時に使用するものである。また、コンテキスト／MPSの16連続判定回路1117は、所定パターンの連続を検出する部分である。

【0054】4ビットシフタ1104は、LPSの領域幅 $Qe(s)$ を左に4ビットシフトして16倍する。シフタ出力は $16 \cdot Qe(s)$ である。減算器1108では $Areg - 16 \cdot Qe(s)$ を計算する。比較器1112は、その値と $0x8000$ を比較する。 $0x8000$ は $Areg$ の初期値 $0x10000$ の $1/2$ である(図2のB点に相当)。

【0055】比較器出力1120は、 $Areg - 16 \cdot Qe(s) \geq 0x8000$ であれば、“1”、そうでなければ“0”とする。ここで、 $Areg - 16 \cdot Qe(s) \geq 0x8000$ であれば、正規化処理は起こらないので、連続処理可能な条件の一つが満たされる。

【0056】一方、コンテキスト／MPS 16連続判定回路1117は、画像データ1119を参照して、同一コンテキストとMPS(s)が16連続しているかどうかを判定する。このために参照する画像データは、図4のa-2～a15、b-2～b17、c-1～c16の各位置にある画像データである。

【0057】本実施例では、これらの画素が全て「白」または全て「黒」を判定する。MPSの符号化を前提とするので、MPS(s)と画素の値が一致していることも判定条件である。判定条件が全て満たされると出力信号1121は“1”、そうでなければ“0”となる。図2に示したように、比較器1112の出力信号1120が、“1”かつ信号1121が“1”のときに、16シンボルの連続処理が可能であるという判断ができる。

【0058】8シンボルの連続処理判断も同様に、3ビットシフタ1103、減算器1107、比較器1111、コンテキスト／MPS 8連続判定回路1116で、連続処理が可能かを判定するための情報を得る。ただし、コンテキスト／MPS

8連続判定回路1116で参照するのは、図4のa-2～a7、b-2～b9、c-1～c8の範囲にある画素データである。4シンボルおよび2シンボルの場合も同様である。これらの異なるシンボル数の判定は同時に(並行して)実行する。

【0059】連続処理シンボル数判定回路1113は、4つの判定結果から連続処理できるシンボル数の最大値を選択する。論理回路は入力信号の組み合わせで容易に実現できる。選択された連続処理シンボル数は、信号1118として算術符号器とコンテキスト生成器1000に出力される。連続処理ができないときは、信号線1118にはゼロを出力する。

【0060】次に、算術符号器1400について説明する。図6は算術符号器1400のブロック図である。

【0061】算術符号化演算は、加算、減算、シフト演算の組み合わせで実行できる。符号化シンボルとMPS(s)はシーケンス制御部1480に入力し、符号化シンボルとMPS(s)が一致するかを比較する。一致した場合はMPSシンボルの符号化、そうでなければLPSシンボルの符号化を実行する。

【0062】MPSの符号化では、レジスタ1440の値 $Areg$ を、 $Areg = Areg - Qe(s)$ とし、 $Areg \geq 0x8000$ であれば処理を終了する。そうでなければ、 $Areg$ と $Qe(s)$ を比較し $Areg < Qe(s)$ であれば、 $A = Qe(s)$ 、 $Creg = Creg + Areg$ とする。

【0063】その後、正規化処理によって $Areg \geq 0x8000$ になるまで左シフトを行う。レジスタ(Creg)1430も同じビット数だけ左シフトする。これらの演算はALU(算術論理演算回路)1450やシフタ1460で実行できる。

【0064】このときレジスタ(Creg)1430から、左にあふれるビットデータをシリアル／パラレル変換器1470で変換したものが符号データバイトとなる。

【0065】連続シンボル数は信号線(図3、1118)を通してシフタ1410に入力し、シフトビット数となる。同じくデータとしてシフタ1410には $Qe(s)$ が入力し、シンボル数分の左シフトを行った値が Qe レジスタ1420に入る。符号化演算は上記の説明で $Qe(s)$ を Qe レジスタの値で置き換える

図7は、逐次処理の場合のMPS処理フローである。LPSの符号化もMPS同様の算術論理演算の組み合わせで実行できる。ステップ310で、“No”であれば、符号は生成されず、ステップ320、330、340を実行する場合だけ符号が出力されることになる。

【0066】以上、符号化について説明した。次に復号化について説明する。但し、符号化と共通する説明は省略する。

【0067】16シンボルの一括復号化に関して図5を使って説明する。図5のコンテキスト／MPS 16連続判定回路1117は、符号化時の判定範囲から復号化シンボルa0～a15を除いた部分のコンテキストの連続性を判定する。

【0068】図8は逐次処理の場合のシンボル復号化フ

ローである。この図を用いてMPSの一括復号条件を説明する。

【0069】このフローで必ずMPS(s)として復号されるのは、Creg < Aregの比較(ステップ410)でYesとなり、更にAreg < 0x8000(ステップ420)でNoとなる場合である。その他のケースでは、Cond_MPS_exchange(ス

Creg < Areg 16*Qe(s) かつ Areg 16*Qe(s) < 0x8000... (1)

コンテキストが16シンボル分同じで、上記の条件が満足されれば、16シンボルをMPS(s)として復元できる。これは図4のシフトレジスタのa0の位置にMPS(s)を設定したまま16ビット左シフトを行うことで実行できる。

【0071】図6においてシンボルの復号化は、外部の符号メモリ(図示せず)から1バイトのデータをレジスタ1430に入力し、Aレジスタ値との比較によって行われる。演算の詳細は省略するが、復元されたシンボルはシーケンス制御部1480からコンテキスト生成器1000に出力される。図5における連続処理シンボル数の判定動作は符号化時とほぼ同様であるが、復号化のときには上記の式(1)の第一条件の判定が加わる。この判定は比較器1131によってCregとAreg 16*Qe(s)の比較をすることによって行われる。

【0072】2シンボル、4シンボル、8シンボルの判定も16シンボルの場合と同様に、またこれらは符号化時同様に同時に実行される。これらの判定結果をもとに連続処理シンボル数判定回路1113で、最大処理シンボル数が選択され算術符号器に出力される。

【0073】以上のように符号化時、復号化時に予め定めた範囲内で連続処理できるシンボルの最大値が選択できる。したがって、画像パターンや解像度の変化に適応して高速化処理が可能となる。また、複数の検出器の各々に連続シンボル数を割り当てて並列に検出させ、各々の出力の中から最大値を選択するというハードウェアによる手法を採用するので、高速な処理が容易に実現されるというメリットもある。

【0074】なお、本実施の形態では、「正規化処理が発生させないこと」を一括処理を行なう条件の一つとしているが、必ず、これを条件としなければならないというものではない。上述のとおり、正規化処理をすると計算が複雑化するので、正規化処理が発生しない範囲で一括処理を行なうのが望ましい、ということである。

【0075】ちなみに、正規化処理が発生しても一括処理をする場合には、以下のような処理をすることになる。

【0076】上述の説明のとおり、一括処理では、コンテキスト(s)に対して、Aregの値をAreg-n*Qe(s)とnシンボル分まとめて計算する。ここで、nシンボルの符号化、復号化中に正規化が発生すると、Qe(s)の値はあらかじめ定められた確率推定器の状態遷移によって更新され、Qe'(s)となる。このような場合には、Areg-n*Qe(s)と単純に計算することはできない。

ステップ430)やCond_LPS_exchange(ステップ450)の処理ルーチン(図示せず)で、復号シンボルDは、MPS(s)にも1 MPS(s)にもなる。

【0070】連続して復号化するシンボル数を16とすると、D = MPS(s)となるルートを16回通る条件は次の通りである。

Creg < Areg 16*Qe(s) かつ Areg 16*Qe(s) < 0x8000... (1)

【0077】したがって、この場合には、正規化が起こるまでのシンボル数を、Areg, Qe(s), 0x8000を使って計算し、それをn1とすると、Aregは、次のように計算されることになる。

Areg = {n1*Qe(s) + (n-n1)*Qe'(s)}

この場合、n1と(n-n1)が2のべき乗であれば計算は容易であるが、一般的にはそうでないので、シフトに代わって乗算器が必要となる。したがって、装置構成に余裕があるのであれば、このような場合にも、本発明を適用して一括処理(連続処理)を行なえる。このことは、以下の実施の形態でも同様である。

【0078】(実施の形態2) 上述の第1の実施の形態では、連続処理可能なシンボル数の判断を複数の可能性について同時に実行した。この場合には、常時、画像の局所状態に応じた最大値の選択ができるが、並列処理のため回路規模がどうしても増大する。

【0079】そこで、回路規模を小さく抑えながら画像の局所状態に適応したシンボル数の決定を可能にしたのが本実施の形態である。

【0080】本実施の形態では、図1の複数シンボル処理判定器1100の構成が、前掲の実施の形態(図5)と異なる。その他は前掲の実施の形態と同じである。

【0081】図9に複数シンボル処理判定器1100のブロック構成を示す。図5の構成と異なり、シフト1150、減算器1151、比較器1152、比較器1159、コンテキスト/MPS連続判定回路1154を一組とする回路と、連続処理シンボル数決定回路1155とで構成されている。

【0082】一括処理の可否を判定し、その結果を実際の処理に反映させるという点では、前掲の実施の形態(図5)と同じであり、また、一括処理の可否を判定する動作も基本的には同じである。但し、図5の場合には、常時、一括処理可能なシンボル数を判定したが、本実施の形態では、所望のシンボル数に対して検出情報に基づき一括処理の可否を判定し、その判定結果を、次のシンボルの符号化に反映させて適応化を図る点(一種の学習効果による適応化を図る点)で異なる。

【0083】図9の連続処理シンボル数決定回路1155は、図10に示すような8つの状態遷移を持つステートマシンである。そして、まず、この連続処理シンボル数決定回路1155が、判定回路1153に判定するべき連続シンボル数を与え、判定回路1153が一括処理が可能か否かを判定し、その結果を連続処理シンボル数決定回路1155に戻し、連続処理シンボル数決定回路1155は、判定結果を次

回の一括処理判定に反映させることで、連続処理シンボル数を適応的に変化させていく。

【0084】以下、具体的に説明する。図10に示すように、連続処理シンボル数決定回路1155の8つの状態の初期値は処理シンボル数が「2」である。

【0085】まず、その初期値「2」がシフト1150、判定回路1153、コンテキスト／MPS連続判定回路1154に入力される。

【0086】この条件で、第1の実施例同様の判定を行い、連続処理可能と判断されるとシンボル数1158には2 = 2¹なので「1」が出力される。記号「 \cdot 」はべき乗である。そうでなければ、「0」が出力される。この判定結果は信号Hit!/!Hit(当たり／外れ)信号として連続処理シンボル数決定回路にフィードバックされる。信号Hit!/!Hitは、今回の例では、シンボル数2で連続処理できる場合に「Hit」、そうでない場合に「!Hit」となる。

【0087】連続処理シンボル数決定回路1155の内部状態は、Hit!/!Hitに応じて遷移し、Hitの場合は次のシンボルの符号化時にシンボル数を大きくして判定する。逆に、!Hitのときは小さくする。なお、遷移状態が最小値あるいは最大値にある場合で、ミスヒットあるいはヒットが発生した場合には、次回もその最小値や最大値を維持するようにする。

【0088】初期値として「2」を設定する場合の動作をまとめると、図11のようになる。すなわち、シンボル数が設定されると(ステップ100)、一括処理の可否が判定され(ステップ110)、一括処理が可能であれば、2シンボル分の一括処理を実行し(ステップ130)、次の一括処理判定における設定値を「4」に変更する(ステップ140)。一方、ステップ110で一括処理が不可と判定された場合には、逐次処理に切り替えて実行し、次の一括処理判定における設定値は「2」に維持される(ステップ120)。

【0089】このような状態遷移によって、全白領域が現れにくいハーフトーンでは小さな値(少ない連続シンボル数)が選択され、全白領域が現れやすい文字画像では大きな値(大きな連続シンボル数)が選択される。この結果として、画像のパターンや局所状態に適応した連続処理シンボル数が得られ、高速な処理が実現される。本実施の形態では、並列に動作する検出器が不要であり、回路規模や消費電流の削減の面でも有利となる。

【0090】図12は、前掲の2つの実施の形態にかかるQM符号／復号化回路を搭載したファクシミリ装置のハードウェア構成を示すブロック図である。

【0091】図示されるように、このファクシミリ装置は、ホストプロセッサ102と、MH／MR／MMR符号／復号化回路103と、解像度変換や拡大・縮小等の処理を行なう画像処理回路104と、QM符号／復号化回路105と、画像ラインメモリ106と、符号メモリ107と、モデム

などの通信インタフェース(電話回線113等を用いた有線伝送のためのインタフェースとして機能する)108と、スキャナ等の画像入力装置111と、プリンタなどの画像記録／表示装置112と、を具備し、各ブロックは内部バス109、110を介して相互に情報の授受を行うことができる。

【0092】本発明の適用によって、画像の種類に柔軟に適応して高速な算術符号化・復号化が実現され、この点でファクシミリ装置(通信装置)の高機能化を図ることができる。

【0093】

【発明の効果】以上説明したように、本発明によれば、各種の画像や画像の局所状態に応じて、予め設定した範囲内で連続処理可能なシンボル数を柔軟に選択することができる。したがって、画像の複雑さに柔軟に対応した高速な符号化・復号化処理を実現できる。

【図面の簡単な説明】

【図1】本発明の実施の形態1にかかる算術符号化装置(算術復号化装置)の構成を示すブロック図

【図2】一括符号化処理を施した場合のオージェクトの分割処理を説明するための図

【図3】一括符号化処理の可否を判定するための処理のフロー図

【図4】実施の形態1に係るコンテキスト生成器の構成を示すブロック図

【図5】実施の形態1に係る複数シンボル処理判定器の構成を示すブロック図

【図6】実施の形態1に係る算術符号器の構成を示すブロック図

【図7】QM-coderのMPS処理(シンボル逐次処理)の手順を示すフロー図

【図8】QM-coderのシンボル復号化処理(シンボル逐次処理)の手順を示すフロー図

【図9】実施の形態2に係る複数シンボル処理判定器の構成を示すブロック図

【図10】実施の形態2に係る複数シンボル処理判定器の状態遷移を示す図

【図11】実施の形態2における一括符号化処理の手順を示すフロー図

【図12】本発明を適用したファクシミリ装置のハードウェア構成例を示すブロック図

【図13】一般的な算術符号器の主要な構成を示す図

【図14】算術符号化の原理を説明するための図

【符号の説明】

1000 コンテキスト生成器

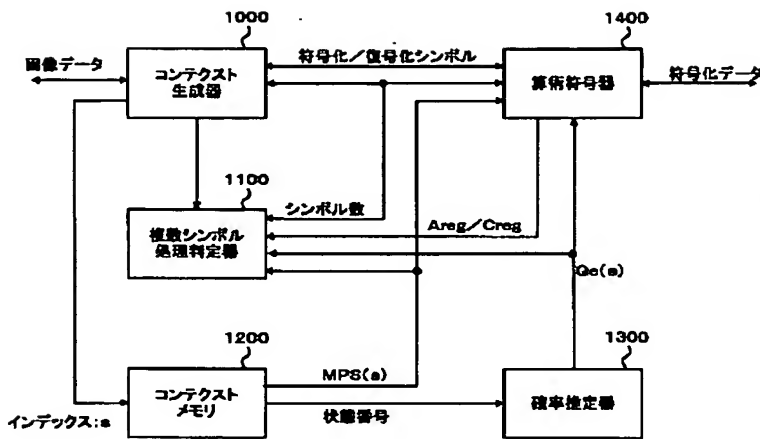
1100 複数シンボル処理判定器

1200 コンテキストメモリ

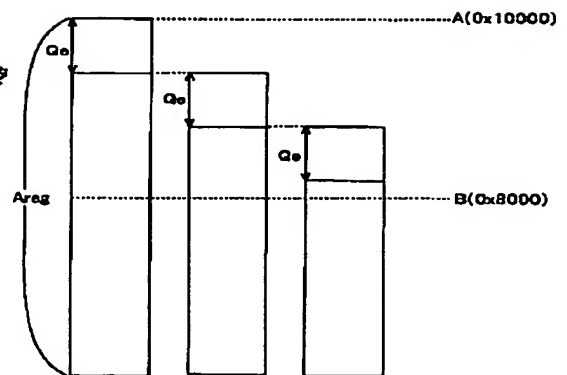
1300 確率推定器

1400 算術符号器

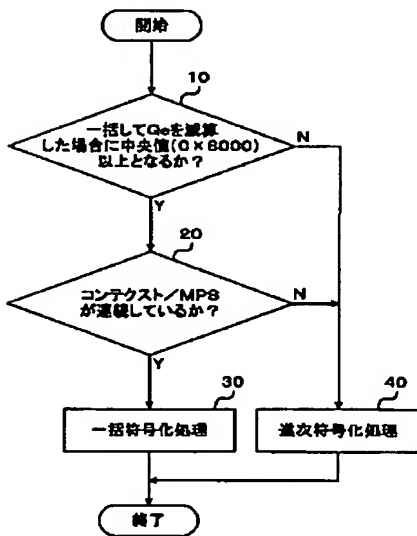
【図1】



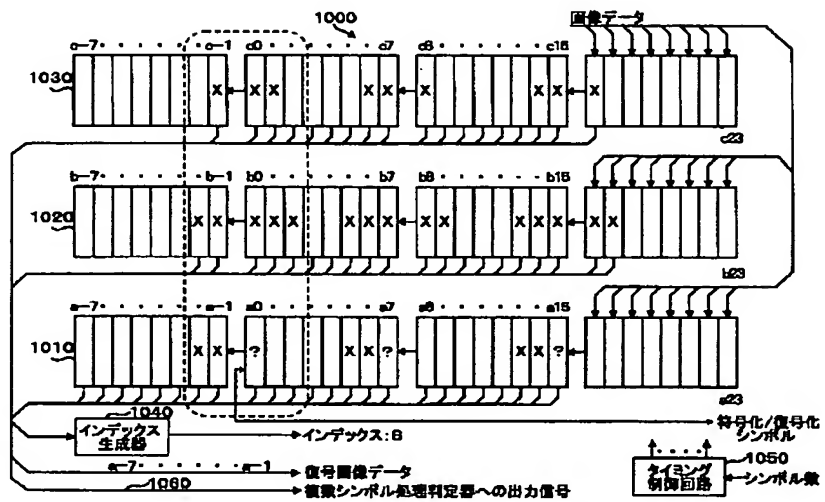
【図2】



【図3】

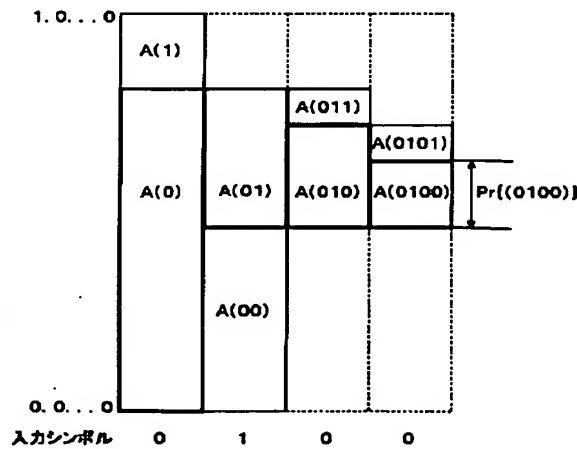
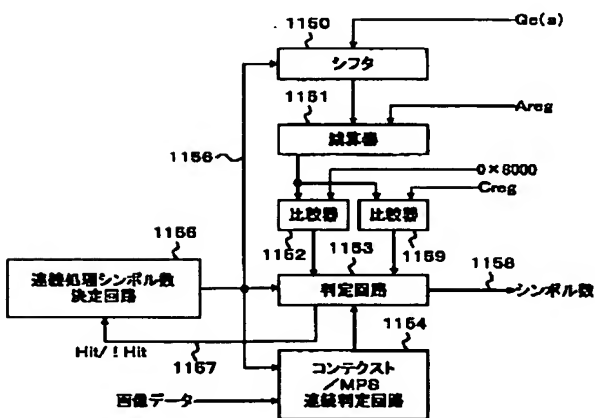


【図4】

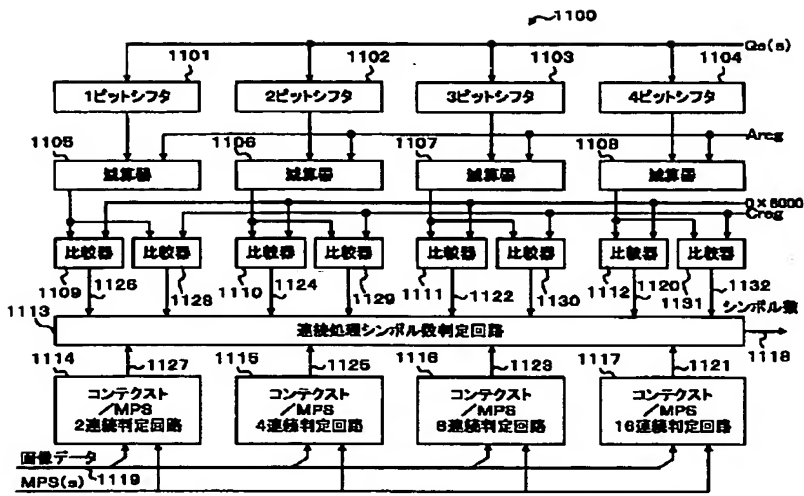


【図14】

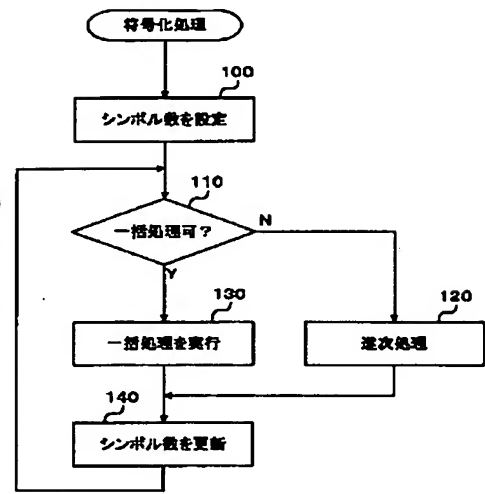
【図9】



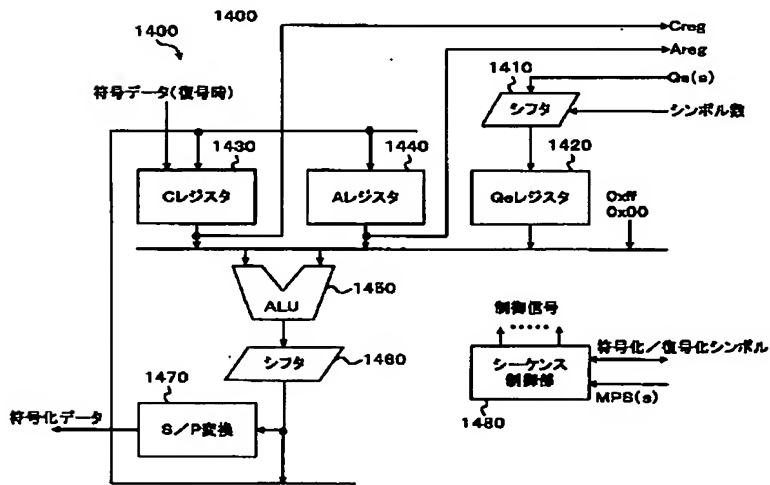
【図5】



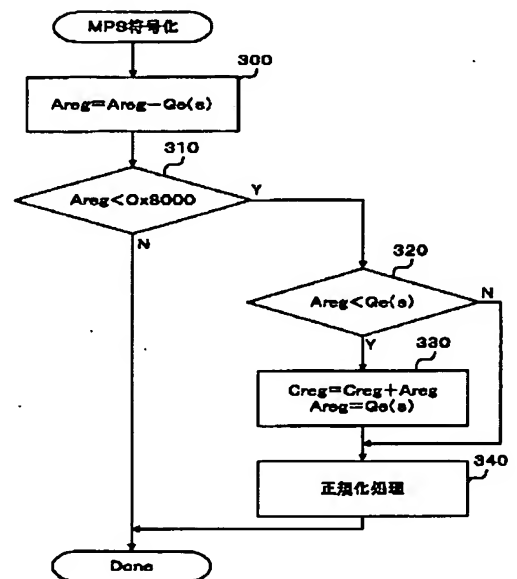
【図11】



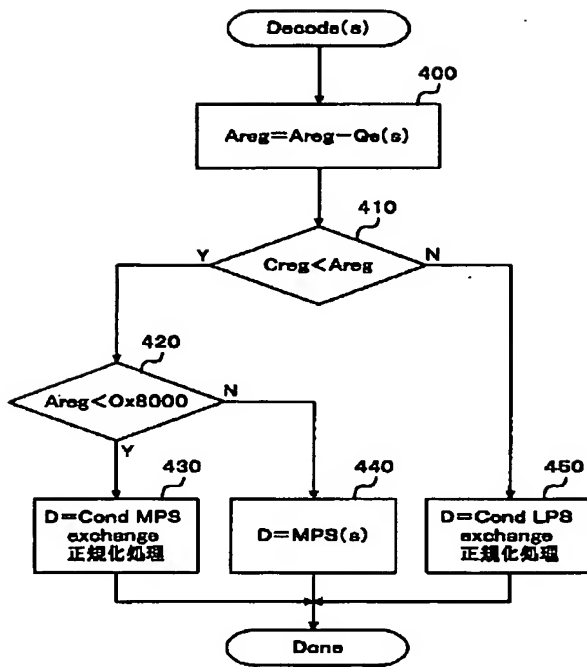
【図6】



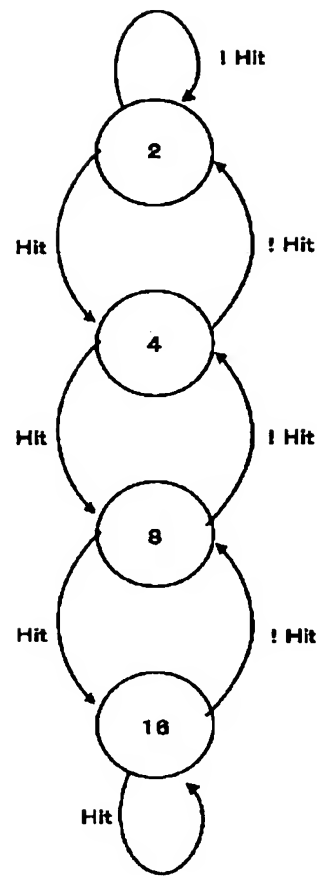
【図7】



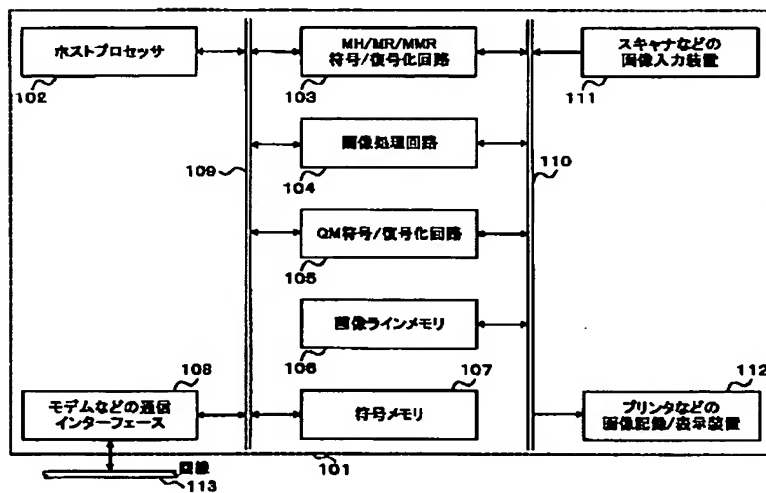
【図8】



【図10】



【図12】



【図13】

